herein provides a pose correction that may not affected by error, interruptions or interference with the video call. In some implementations, a depth image of the user could be used to help perform pose correction of the user and to fill in any areas where the depth map acquired by the user device may have hidden areas or gaps.

Some implementations can ignore objects briefly blocking the face of the user, predict where the face would be using historical data, perform pose correction based on the prediction, and then restore to normal pose correction operation function when the object moves. In some implementations, when an object is in front of the face of a user, the pose correction can fade out the correction and not rely on historical data.

In some implementations, the depth image can include a vector of floating point values, a projection (original or virtual camera) can include a 4×4 matrix that maps the cones of camera to a cube including a mapping of position and orientation (forward, up, and back).

In some implementations, when two or more faces are present in video frames, an implementation can be configured to separate the two (or more) users and treat as two (or more) foregrounds. In some implementations, if two (or more) users are detected then the pose correction can be backed out.

In some implementations, objects on the face can be accommodated by the pose correction technique described herein. For example, glasses and hats can be handled. Theoretically, any object on the face or in front of the face can be an issue for pose correction. The objects on the face of a user can be handled by greater smoothing of depth data and, if holes (e.g., areas of face that are occluded but would be visible after correction) are detected, then the system backs out the pose correction. Backing out the pose correction can include reducing the amount of correction being applied. For example, a maximum limit can be initially 20 degrees. To back out the pose correction, the 20 degrees is gradually reduced to 0 degrees. To smoothly re-apply the effect, the 0 degrees can be gradually increased back to 20.

In some implementations, the system can adjust correction angle based on movement of the device (e.g., to compensate for movement of the device during the video call). In some implementations, if holes (e.g., areas of face that are occluded but would be visible after correction) are detected then pose correction is not applied. Implementations can be used to correct poses in security cameras or other cameras. Some implementations can be used to create a parallax effect based on the user on the other side of the call.

FIG. **7** is a block diagram of an example device **700** which may be used to implement one or more features described herein. In one example, device **700** may be used to implement a user device, e.g., user devices **104**, **302**, **402**, shown in FIGS. **1**, **3**, and **4**, respectively. Alternatively, device **700** can implement a server device. In some implementations, device **700** may be used to implement a client device, a server device, or a combination of the above. Device **700** can be any suitable computer system, server, or other electronic or hardware device for implementing one or more of the systems and/or methods described above.

One or more methods described herein (e.g., **500** and/or **600**) can be run in a standalone program that can be executed on any type of computing device, a program run on a web browser, a mobile application ("app") run on a mobile computing device (e.g., cell phone, smart phone, tablet computer, wearable device (wristwatch, armband, jewelry, headwear, virtual reality goggles or glasses, augmented reality goggles or glasses, head mounted display, etc.), laptop computer, etc.).

In one example, a client/server architecture can be used, e.g., a mobile computing device (as a client device) sends user input data to a server device and receives from the server the final output data for output (e.g., for display). In another example, all computations can be performed within the mobile app (and/or other apps) on the mobile computing device. In another example, computations can be split between the mobile computing device and one or more server devices.

In some implementations, device **700** includes a processor **702**, a memory **704**, and I/O interface **706**. Processor **702** can be one or more processors and/or processing circuits to execute program code and control basic operations of the device **400**. A "processor" includes any suitable hardware system, mechanism or component that processes data, signals or other information. A processor may include a system with a general-purpose central processing unit (CPU) with one or more cores (e.g., in a single-core, dual-core, or multi-core configuration), multiple processing units (e.g., in a multiprocessor configuration), a graphics processing unit (GPU), a field-programmable gate array (FPGA), an application-specific integrated circuit (ASIC), a complex programmable logic device (CPLD), dedicated circuitry for achieving functionality, a special-purpose processor to implement neural network model-based processing, neural circuits, processors optimized for matrix computations (e.g., matrix multiplication), or other systems.

In some implementations, processor **702** may include one or more co-processors that implement neural-network processing. In some implementations, processor **702** may be a processor that processes data to produce probabilistic output, e.g., the output produced by processor **702** may be imprecise or may be accurate within a range from an expected output. Processing need not be limited to a particular geographic location, or have temporal limitations. For example, a processor may perform its functions in "real-time," "offline," in a "batch mode," etc. Portions of processing may be performed at different times and at different locations, by different (or the same) processing systems. A computer may be any processor in communication with a memory.

Memory **704** is typically provided in device **700** for access by the processor **702**, and may be any suitable processor-readable storage medium, such as random access memory (RAM), read-only memory (ROM), Electrically Erasable Read-only Memory (EEPROM), Flash memory, etc., suitable for storing instructions for execution by the processor, and located separate from processor **702** and/or integrated therewith. Memory **704** can store software operating on the device **700** by the processor **702**, including an operating system **708**, pose correction application **730**, other applications **712**, and application data **714**. Other applications **712** may include applications such as a data display engine, web hosting engine, image display engine, notification engine, social networking engine, etc. In some implementations, the pose correction application **730** and other applications **712** can each include instructions that enable processor **702** to perform functions described herein, e.g., some or all of the methods of FIGS. **5** and/or **6**. Any of software in memory **704** can alternatively be stored on any other suitable storage location or computer-readable medium. Memory **704** and any other type of storage (magnetic disk, optical disk, magnetic tape, or other tangible media) can be considered "storage" or "storage devices."